

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
4 août 2005 (04.08.2005)

PCT

(10) Numéro de publication internationale
WO 2005/071536 A2

(51) Classification internationale des brevets⁷ : **G06F 9/44**

(21) Numéro de la demande internationale :
PCT/EP2004/053527

(22) Date de dépôt international :
16 décembre 2004 (16.12.2004)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
03/15354 24 décembre 2003 (24.12.2003) FR

(71) Déposant (pour tous les États désignés sauf US)
: **THALES** [FR/FR]; 45, rue de Villiers, F-92200
Neuilly-sur-Seine (FR).

(72) Inventeurs; et

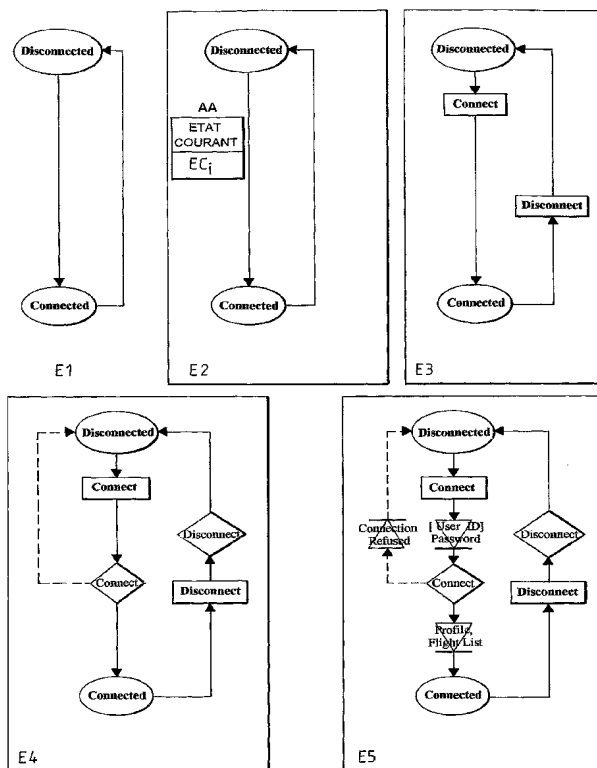
(75) Inventeurs/Déposants (pour US seulement) : **FAURE, David** [FR/FR]; Thales Intellectual Property, 31-33, avenue Aristide Briand, F-94117 Arcueil Cedex (FR). **GRISVARD, Olivier** [FR/FR]; Thales Intellectual Property, 31-33, avenue Aristide Briand, F-94117 Arcueil Cedex (FR). **PRAUD, Sébastien** [FR/FR]; Thales Intellectual Property, 31-33, avenue Aristide Briand, F-94117 Arcueil Cedex (FR). **SEDOGBO, Célestin** [FR/FR]; Thales Intellectual Property, 31-33, avenue Aristide Briand, F-94117 Arcueil Cedex (FR).

(74) Mandataires : **CHAVERNEFF, Vladimir** etc.; Thales Intellectual Property, 31-33, avenue Aristide Briand, F-94117 Arcueil Cedex (FR).

[Suite sur la page suivante]

(54) Title: METHOD FOR IMPROVING A WORKING MODEL FOR THE MANAGEMENT OF THE MAN-MACHINE INTERACTION

(54) Titre : PROCEDE D'AUGMENTATION D'UN MODELE DE TÂCHE POUR PERMETTRE LA GESTION DE L'INTERACTION HOMME-MACHINE



AA ... COURANT STATE EC₁

(57) Abstract: The invention relates to a method characterised in that an existent working model, as may be provided by an expert, may be improved with the actual status of the user within the task thereof, the actions permitting a change of status for the user are described and the interaction to be carried out with the user to manage an event for an event arising for a status of the user are described. The list of conditions necessary to start the interaction are advantageously added for each interaction procedure and, after each interaction procedure, the values which said interaction, according to the result of said interaction, should provide are added and which also should be provided to the user by return.

(57) Abrégé : Le procédé conforme à l'invention est caractérisé en ce qu'à partir d'un modèle de tâche existant, qui peut être fourni par un expert, on l'augmente avec l'état courant de l'utilisateur dans sa tâche, on décrit les événements permettant un changement d'état de l'utilisateur, on décrit pour un événement survenu lors d'un état de l'utilisateur l'interaction à effectuer avec l'utilisateur pour gérer cet événement. De façon avantageuse, on ajoute avant chaque procédure d'interaction la liste des contraintes nécessaires pour déclencher l'interaction, et on ajoute après chaque procédure d'interaction les valeurs que doit fournir cette interaction selon le résultat de l'interaction et qui doivent être présentées à l'utilisateur en retour.

WO 2005/071536 A2



- (81) **États désignés** (*sauf indication contraire, pour tout titre de protection nationale disponible*) : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) **États désignés** (*sauf indication contraire, pour tout titre de protection régionale disponible*) : ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Publiée :

— *sans rapport de recherche internationale, sera republiée dès réception de ce rapport*

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

Procédé d'augmentation d'un modèle de tâche pour permettre la gestion de l'interaction homme-machine

La présente invention se rapporte à un procédé d'augmentation d'un modèle de tâche pour permettre la gestion de l'interaction homme-machine.

Dans les systèmes actuels disposant d'une interface avec un humain, le modèle de la tâche de l'utilisateur est enfoui au sein de l'interface. La tâche que l'utilisateur doit effectuer n'est pas explicitement définie, mais se retrouve implicitement dans la conception de l'interface. Les concepteurs d'interfaces définissent, selon les spécifications, les enchaînements plus ou moins stricts qui sont autorisés à l'utilisateur. Ces enchaînements définissent implicitement son modèle de tâche.

10 Cette absence de définition explicite du modèle de tâche d'un utilisateur a pour conséquence une forte staticité de la tâche de l'utilisateur. Dans le cas où la mission de l'utilisateur change ou si l'interface doit être déployée pour une classe d'utilisateurs ayant une mission partiellement différente, des modifications sont nécessaires sur l'interface. Certaines interfaces sont conçues en prenant en compte
15 cette optique de modifications et peuvent être configurées pour être adaptées à une tâche différente. Cette pratique impose une forte charge de travail pour concevoir une interface suffisamment ouverte pour permettre différentes configurations. De plus, la configuration de l'interface pour une nouvelle tâche reste complexe dans la mesure où la configuration nécessite le travail conjoint d'un expert du domaine pour
20 spécifier la nouvelle tâche et d'un concepteur d'interface. Cette pratique, déjà relativement coûteuse, devient particulièrement difficile à mettre en place si la tâche de l'utilisateur est susceptible d'évoluer au cours d'une session, ce qui oblige à modifier l'application renfermant le modèle de tâche.

La présente invention a pour objet un procédé d'augmentation d'un modèle
25 de tâche permettant la gestion en temps réel de l'interaction homme-machine sans avoir à modifier l'application liée à ce modèle de tâche, en particulier lorsque l'on fait évoluer les modèles de tâches, l'interaction pouvant être multi-utilisateurs.

Le procédé conforme à l'invention est caractérisé en ce qu'à partir d'un modèle de tâche existant, on l'augmente avec l'état courant de l'utilisateur dans sa tâche, on décrit les événements permettant un changement d'état de l'utilisateur, et on décrit pour un événement survenu lors d'un état de l'utilisateur l'interaction à effectuer avec l'utilisateur pour gérer cet événement. De façon avantageuse, on ajoute avant chaque procédure d'interaction la liste des contraintes nécessaires pour déclencher l'interaction, et, également de façon avantageuse, on ajoute après chaque procédure d'interaction les valeurs que doit fournir cette interaction selon le résultat de l'interaction et qui doivent être présentées à l'utilisateur en retour.

La présente invention sera mieux comprise à la lecture de la description détaillée d'un mode de réalisation, pris à titre d'exemple non limitatif et illustré par le dessin annexé, sur lequel :

- la figure 1 est un diagramme simplifié explicitant différentes étapes de la mise en œuvre du procédé de l'invention dans une application,
- la figure 2 est un bloc-diagramme simplifié de l'architecture d'un gestionnaire de tâches conforme à l'invention, et
- la figure 3 est un diagramme simplifié du déroulement des tâches du gestionnaire de tâches de l'invention.

Le procédé de l'invention consiste essentiellement à décharger les concepteurs d'interfaces homme-machine de la question de la tâche de l'utilisateur. Les concepteurs n'auront qu'à se concentrer sur les fonctionnalités d'interface dont l'utilisateur a besoin sans se préoccuper des enchaînements qu'il aura à réaliser pour mener à bien sa tâche.

L'invention part d'un modèle de tâche externe existant et l'augmente de telle sorte qu'il puisse être utilisé pour piloter l'interaction. De plus, ce modèle sert de base de spécification aux concepteurs d'interfaces pour connaître les services nécessaires aux utilisateurs.

Enfin, ce modèle de tâche externe est utilisé par un gestionnaire de tâches qui est capable de prendre en compte en temps réel tout changement dans le modèle de tâche lors de l'exécution de la tâche.

L'invention décrite ici précise et met en œuvre des réponses aux problèmes posés par le besoin de développement et de configuration des interfaces suite à des modifications de tâche de l'utilisateur ou suite à une demande de déploiement à l'intention d'autres utilisateurs avec une tâche différente. En ce sens, cette invention
5 a des répercussions sur toutes les catégories de systèmes où l'humain tient une place prépondérante, et ceci dans toutes sortes de domaines d'application, comme par exemple :

- La défense : systèmes d'armes, de commandement, de simulation et d'entraînement ;
- 10 • Les transports : systèmes de pilotage, de supervision, de réservation ;
- Les communications : téléphonie, radio, télévision, Internet ;
- La vie quotidienne : systèmes électroménagers, véhicules individuels, informatique domestique omniprésente et diffuse (« *ubiquitous computing* » en anglais) ;
- 15 • Les services : systèmes bancaires, commerce électronique, assistance technique ;
- La santé : systèmes hospitaliers, secours opérationnel ;
- Etc.

La présente invention propose un nouveau procédé pour augmenter un
20 modèle de tâche existant afin de permettre une gestion en temps réel de l'interaction pilotée par le modèle de tâche. Son principe fondamental est d'utiliser un modèle existant de tâche des utilisateurs et de l'augmenter avec les connaissances nécessaires pour le pilotage de l'interaction. Plus précisément, le modèle de tâche d'un opérateur comporte, en plus de l'enchaînement des tâches à réaliser, les événements pouvant
25 survenir, la description des interactions à effectuer, les conditions de déclenchement nécessaires et les informations à fournir en retour. Ce modèle de tâche augmenté permet de piloter l'interaction en temps réel grâce à un composant supplémentaire, le gestionnaire de tâches, qui fournit les services d'accès au modèle de tâche. Le modèle de tâche en tant que connaissance externe à l'interaction est dynamique. De

ce fait, le gestionnaire de tâches autorise une évolution du modèle de tâche au cours du temps avec répercussion en temps réel sur l'interaction avec l'utilisateur.

Habituellement, un modèle de tâche décrit les différentes étapes qu'un utilisateur peut ou doit effectuer pour mener à bien sa tâche globale. La tâche de l'utilisateur doit être décrite sous forme d'un enchaînement alternatif, séquentiel ou en parallèle de sous-tâches avec une mention particulière pour l'état initial. Selon l'invention, les sous-tâches effectuées par l'utilisateur correspondent à autant d'états dans lesquels se trouve l'utilisateur (en état d'effectuer une sous-tâche).

La première étape du procédé de l'invention part d'un modèle de tâche existant fourni par un expert, et elle consiste à augmenter cette représentation avec l'état courant de l'utilisateur dans sa tâche. Cette connaissance supplémentaire permet de suivre l'utilisateur dans sa tâche.

La deuxième étape consiste à décrire les événements permettant un changement d'état de l'utilisateur (fin d'une sous-tâche pour en débiter une autre). Ces événements peuvent être des événements à l'initiative de l'utilisateur ou des événements externes survenus dans l'environnement. L'environnement est défini ici comme tout excepté l'utilisateur : la ou les applications, les autres utilisateurs, des capteurs,... Ce procédé ajoute ainsi sur chaque transition entre deux états de l'utilisateur la liste d'un ou de plusieurs événements déclenchant le changement d'état. Les événements inscrits dans le modèle de tâche peuvent être de granularité plus ou moins fine selon les besoins. Selon une caractéristique avantageuse de l'invention, on utilise un module externe qui fournit les événements (on l'« utilise » dans le sens d'un client utilisant un fournisseur). Ce module est n'importe quelle sorte d'interface homme-machine (informatique ou non). Il peut, par exemple, s'agir d'un capteur de poids sur un siège qui fournit un événement « assis ». Ce module fournit une abstraction des actions de l'utilisateur sous forme d'événements de haut niveau. Cette abstraction permet d'utiliser le même événement dans le modèle de tâche quelle que soit la manière dont l'utilisateur a interagi avec le système (vocalement, graphiquement, par geste,...). Il est également possible d'utiliser des événements de bas niveau dans le modèle de tâche. La distinction entre événements de bas niveau et de haut niveau étant faite ici dans le sens où celui de haut niveau est

la généralisation d'un ou plusieurs événements de bas niveau. Par exemple, il peut être avantageux, dans un modèle de tâche, de déclencher une action sur l'événement « Select » (de haut niveau), que ce soit une sélection à la souris, avec un joystick ou une désignation par geste. D'un autre côté, il peut être avantageux de pouvoir définir

5 trois actions différentes pour ces trois événements (bas niveau). On peut utiliser une interface (quelle qu'elle soit) qui puisse renvoyer un événement générique (Select) dans certains contextes et des événements spécifiques dans d'autres contextes.

La troisième étape consiste à décrire, pour un événement survenu lors d'un état de l'utilisateur, l'interaction à effectuer avec l'utilisateur pour gérer cet

10 événement. L'interaction est définie comme une procédure à dérouler (interrogation d'applications, calcul d'une valeur, vérification d'une donnée,...). En fait, le modèle de tâche peut faire référence à plusieurs applications différentes. Ainsi, l'utilisateur a une seule et même interface, et ce sont alors les procédures d'interaction qui se chargent d'accéder aux applications nécessaires. Chaque procédure d'interaction doit

15 fournir un résultat permettant de décider du prochain état de l'utilisateur. Pour chaque résultat possible de la procédure d'interaction, une transition étiquetée par le résultat permet d'atteindre un nouvel état de l'utilisateur. Par exemple, dans le cas où l'utilisateur est dans l'état « déconnecté » et qu'il déclenche un événement de « connexion », la procédure d'interaction, après avoir interrogé l'application, fournit

20 une réponse polaire (oui/non). Si la réponse est positive, l'utilisateur accède à l'état « connecté », sinon il reste dans l'état « déconnecté », comme décrit dans le modèle de tâche augmenté.

La quatrième étape consiste à ajouter avant chaque procédure d'interaction la liste des contraintes nécessaires pour déclencher l'interaction. Ces contraintes

25 peuvent être la fourniture de paramètres, la vérification d'une valeur,... (Dans l'exemple précédent, la contrainte pourrait être la fourniture d'un identifiant et d'un mot de passe.)

La dernière étape consiste à ajouter après chaque procédure d'interaction les valeurs que doit fournir cette interaction selon le résultat de l'interaction et qui

30 doivent être présentées à l'utilisateur en retour. (Dans l'exemple précédent, la sortie

de la procédure d'interaction permettant la connexion pourrait être le « mot du jour ».)

Le procédé de l'invention identifie clairement les actions à effectuer par l'utilisateur ainsi que leur séquencement, les événements pouvant être déclenchés ou
5 perçus par l'utilisateur ainsi que les interactions (au sens de la tâche) avec l'environnement.

Un modèle de tâche ainsi augmenté peut se présenter sous forme de graphe. Le formalisme utilisé pour ce graphe doit offrir une puissance d'expression suffisante pour exprimer les états (sous-tâches) de l'utilisateur, les contraintes à vérifier pour
10 l'exécution d'une procédure d'interaction, les valeurs fournies par une procédure d'interaction, les événements pouvant survenir et la manière de les prendre en compte. Il est aussi nécessaire que le formalisme permette la représentation de l'interruption de tâche, la reprise après interruption, etc.

La gestion en temps réel d'un tel modèle de tâche est effectuée par le
15 gestionnaire de tâches encapsulant l'accès aux modèles de tâches. Il se comporte comme un fournisseur de services qui seront décrits ci-après. Le gestionnaire est à même de gérer plusieurs modèles de tâches différents et plusieurs utilisateurs en parallèle. La multiplicité des modèles de tâches gérés par un même gestionnaire permet la collaboration entre plusieurs utilisateurs ayant des tâches potentiellement
20 différentes. Le gestionnaire de tâches prend en compte l'évolutivité dans le temps d'un modèle de tâche en offrant des services d'accès pour des composants qui pourraient altérer les modèles.

Afin de faciliter l'écriture et la relecture de ces modèles de tâches, des outils de saisie et de relecture sont mis en œuvre. Ces outils permettent à un expert de
25 pouvoir facilement exprimer ou vérifier la tâche d'un ou de plusieurs utilisateurs, de préciser les différentes synchronisations dans le cas d'un travail collaboratif et de vérifier la cohérence des modèles.

Les principaux avantages du procédé de l'invention sont les suivants. D'abord, la présente invention consiste à augmenter le modèle de tâche avec de
30 nouvelles connaissances. Ce modèle de tâche augmenté permet de piloter l'interaction homme-machine en temps réel. Les concepteurs des modules gérant

l'interaction avec l'utilisateur n'ont plus à se soucier de la manière de gérer l'interaction, ces informations étant localisées au sein du modèle de tâche et accessibles via le gestionnaire de tâches.

L'invention présente aussi l'avantage de mémoriser le contexte de tâche d'un
5 utilisateur et est ainsi à même d'aider à maintenir une cohérence dans sa mission, que ce soit en cas d'interruption volontaire (fin de session,...) ou non volontaire (panne impliquant une interruption ou un changement de poste de travail,...).

Un autre avantage est la génération automatique d'interfaces. Le modèle de tâche décrivant l'interaction nécessaire à l'utilisateur selon le point d'avancement de
10 sa tâche, il devient possible de mettre en place un système de génération automatique d'interfaces au vol. Le modèle de tâche fournit les informations nécessaires sous forme d'états pouvant être atteints et d'événements déclenchables selon les contraintes, en fonction de l'état courant de l'utilisateur.

Un autre avantage repose sur la modification en temps réel du modèle de
15 tâche. L'interaction avec l'utilisateur peut être modifiée en temps réel dans le cas où des contraintes externes l'exigent, comme par exemple une modification des niveaux de sécurité. L'interface de l'utilisateur peut alors être modifiée en temps réel.

L'invention décrite ici, en plus des avantages cités précédemment, ouvre la voie à de nouvelles possibilités. Parmi ces possibilités on peut citer l'apprentissage
20 par un module externe. Ce module doit recevoir tous les changements d'état de l'utilisateur en temps réel et doit être capable d'apprendre son comportement typique. L'apprentissage peut se faire en temps réel ou en différé. à partir d'un modèle de tâche (ce que devrait faire l'utilisateur) de l'activité d'un utilisateur (ce qu'il fait réellement), pouvant ainsi permettre d'influer sur son modèle de tâche en
25 temps réel. Le module d'apprentissage est alors à même d'altérer le modèle de tâche pour simplifier la tâche, pour optimiser son déroulement ou pour s'adapter à la manière de travailler de chaque utilisateur. Le procédé permet la prise en compte en temps réel de ces altérations.

Un autre avantage est de permettre la mise en place aisée de techniques de
30 planification des tâches de l'utilisateur. Cette planification permet de prévoir ses

actions et d'anticiper ses demandes (chargement par anticipation de modules logiciels, téléchargement anticipé de données longues à obtenir,...).

Selon une autre caractéristique de l'invention, on applique la séparation entre l'application et l'interface homme-machine décrite dans la demande de brevet
5 français 02 12012. L'interface est alors développée sous forme de services d'interaction selon les spécifications du modèle de tâche. L'interface ne comporte alors plus le modèle de tâche enfoui de l'utilisateur. Avant, l'interface définissait que telle fenêtre présentait telles données et permettait l'accès à telles autres fenêtres (états). Désormais, selon l'invention, les développeurs d'interfaces connaissent les
10 fenêtres à développer en consultant le modèle de tâche. Pour chaque état, il faut une fenêtre permettant d'afficher les informations à présenter (diode précédant l'état, en référence à la figure 1, décrite ci-dessous) et de proposer des fonctionnalités pour déclencher les événements de sortie (des boutons, par exemple). Ainsi, le développeur d'interfaces ne développe plus l'interface complète, mais une liste de
15 services d'interface pour les états. Pour l'adaptation automatique de l'interface en temps réel, on peut envisager des systèmes où, selon l'état dans lequel va arriver l'utilisateur, connaissant les données à lui présenter et les fonctions nécessaires (boutons), on choisit le meilleur service d'interface dans une liste. Il peut donc y avoir trois interfaces permettant de présenter la liste des vols et d'accéder à la suite
20 du modèle de tâche. La première vocale, la seconde pour un grand écran et la troisième pour un PDA (assistant personnel). Le gestionnaire de tâches fournit la connaissance définissant les besoins en termes d'interface (présenter une liste de vol et les événements déclenchables), et c'est un autre module qui, selon le contexte de l'utilisateur, appelle tel ou tel service d'interface. Cette caractéristique est très
25 avantageuse dans la mesure où plusieurs services d'interface peuvent exister en parallèle et être choisis selon le contexte. Cette adaptation permet à l'utilisateur d'effectuer une même tâche sur des terminaux différents, avec des modalités différentes,...

De plus, le choix des procédures d'interaction étant défini au sein du modèle
30 de tâche, l'approche du développement de l'interface du point de vue des services la rend indépendante de l'application sous-jacente. Un changement d'application

continuant à offrir des services similaires, impliquera seulement une mise à jour des procédures d'interaction du modèle de tâche sans altération nécessaire sur l'interface.

En plus des adaptations précédentes, le développement d'une interface sous forme de services grâce au modèle de tâche permet de faire varier le modèle de tâche
5 selon l'utilisateur sans impliquer de développement supplémentaire au niveau de l'interface.

Le modèle de tâche augmenté devient aussi la spécification permettant la conception et le développement des interfaces et permet de fédérer les développeurs de l'application avec les développeurs d'interfaces. Pour les concepteurs d'interfaces,
10 ce modèle de tâche décrit les interactions qui seront nécessaires à l'utilisateur pour effectuer sa tâche. Ce modèle, en tant que spécification d'interaction, garantit une interface répondant aux besoins des utilisateurs. De plus, le modèle de tâche définit les procédures d'interaction (généralement des ensembles d'appels de services applicatifs) à déclencher lors d'une demande de changement d'état de la part de
15 l'utilisateur. Ceci permet aux développeurs d'applications de connaître la liste des services nécessaires à l'utilisateur. Ce formalisme permet aux concepteurs d'interface de s'abstraire de tous les services applicatifs, car la description des services applicatifs à déclencher est précisée dans le modèle de tâche.

Une réalisation possible de la présente invention est l'augmentation du
20 modèle de tâche d'un utilisateur. La connaissance fournie en entrée du procédé est un modèle de tâche d'un utilisateur. Ce procédé a permis l'augmentation du modèle de tâche en collaboration avec les experts afin d'obtenir une version étendue dont un exemple partiel est donné ci-après.

Afin de gérer au mieux ce modèle de tâche, un mode de réalisation consiste à
25 mettre en place un « gestionnaire de tâches ». Ce dernier a pour objectif de fournir les services d'accès à la tâche pour les autres modules logiciels qui en ont besoin. Le modèle de tâche repose, dans le présent exemple, sur le format XML, aisé à percevoir pour les experts qui devront l'exprimer et pour les concepteurs d'interfaces qui devront l'utiliser. Le gestionnaire de tâche fournit l'accès à plusieurs modèles de
30 tâches différents pour plusieurs utilisateurs, en parallèle et en temps réel.

Le modèle de tâche décrit les différents états de l'utilisateur au sein de sa tâche et décrit les actions qu'il peut entreprendre pour changer d'état. Ce formalisme définit l'état de départ d'un utilisateur qui débute sa tâche. Pour chacun des états de l'utilisateur, un certain nombre d'actions possibles (potentiellement une seule) peuvent le conduire dans un nouvel état en déclenchant une procédure d'interaction (généralement un appel à un ou plusieurs services applicatifs). Les transitions présentes dans ce formalisme peuvent comporter des contraintes. La nécessité de disposer d'une donnée particulière pour autoriser le déclenchement d'une action est une forme de contrainte. Les modèles de tâches de plusieurs utilisateurs peuvent être liés entre eux de manière à pouvoir prendre en compte les problèmes de travail collaboratif.

Selon le formalisme de l'invention, les états de l'utilisateur sont décrits par des ovales, les procédures d'interaction par des losanges et les paramètres nécessaires pour déclencher une procédure par des diodes. Les informations fournies par une procédure d'interaction sont aussi décrites sous forme de diodes et indiquent aux développeurs d'interfaces les informations résultant de la précédente procédure d'interaction qu'il faudra présenter à l'utilisateur.

Dans le cas où il existe une hiérarchie des classes d'utilisateurs en termes de tâche où certains utilisateurs ne disposent que d'une tâche partielle à réaliser par rapport à d'autres ayant la tâche complète, le modèle de tâche permet de spécifier la hiérarchie de modèles de tâches afin de faciliter la spécialisation ou la généralisation des modèles.

Le gestionnaire de tâches de l'invention se présente sous forme d'un module externe et temps réel fournissant l'accès aux modèles de tâche dont il a la charge. Il fournit tous les services d'accès nécessaires, comme l'initialisation d'un modèle de tâche lors de la connexion d'un nouvel utilisateur, l'identification du modèle de tâche selon la classe de l'utilisateur, la fourniture de la prochaine procédure d'interaction en fonction d'un événement, la fourniture du prochain état en fonction du résultat d'une procédure d'interaction. Ce module est capable de gérer en parallèle plusieurs utilisateurs en mémorisant le contexte de chacun (son processus de fonctionnement est décrit en référence à la figure 3). La sécurité de fonctionnement de la gestion de

tâche doit être assurée, par exemple avec une vérification des actions de l'utilisateur vis-à-vis de sa tâche. Le gestionnaire de tâches fournit aussi les services d'accès en écriture aux différents modèles de tâches avec sauvegarde des modèles altérés. Le gestionnaire de tâches peut utiliser un module de stockage externe afin d'accéder aux
5 modèles de tâches des utilisateurs.

On a représenté en figure 1 cinq étapes principales du procédé d'augmentation de tâche, conformément à l'invention. Une flèche pleine correspond à une réponse positive et une flèche pointillée correspond à une réponse négative. Dans cette figure, la première étape E1 représente le modèle de tâche initial. On y a
10 figuré dans des ellipses deux états successifs de l'utilisateur, à savoir "Disconnected" (déconnecté, ou plus précisément, non encore connecté à un appareil, par exemple un microordinateur), et "Connected" (connecté à cet appareil). On parle d'état initial lorsqu'il s'agit du premier état dans lequel se trouve un utilisateur qui commence sa tâche.

15 À l'étape suivante E2, on ajoute l'état courant de l'utilisateur à l'aide d'une « balise » stockée par le gestionnaire de tâches. Cet état courant se présente, par exemple, comme une variable ec_i , (i pouvant prendre les valeurs 1 à n pour n états possibles de l'utilisateur).

Le procédé de l'invention consiste ensuite (étape E3) à introduire entre ces
20 deux états la description d'un événement (figurée dans un rectangle), qui est ici "Connect" (demande de connexion), et qui permet de faire passer l'utilisateur du premier état vers le deuxième. Inversement, on introduit la description d'un autre événement, "Disconnect", qui permet de faire passer l'utilisateur du deuxième vers le premier état.

25 À l'étape suivante E4, on ajoute les procédures d'interaction (figurées chacune dans un losange. Ces procédures sont la connexion de l'utilisateur à la machine (« connect ») et la déconnexion (« disconnect »).

Après la description d'événement "Connect", on introduit (étape E5) les paramètres régissant le déclenchement de l'action prévue (figurés par une diode),
30 c'est-à-dire le déclenchement du processus de connexion de l'utilisateur à la machine.

Dans le cas présent, il y a deux tels paramètres, "User ID" (identité de l'utilisateur) et "Password" (mot de passe). la description de la procédure d'interaction avec l'utilisateur, et dans le cas présent, il s'agit simplement de la procédure de connexion "Connect" citée ci-dessus. On remarquera que pour faire passer l'utilisateur du

5 deuxième vers le premier état, il n'y a pas besoin de faire intervenir de paramètres, la déconnexion devant se faire inconditionnellement. Lors de cette procédure de connexion, il y a, par exemple, vérification de l'exactitude des paramètres fournis par l'utilisateur à l'aide d'un clavier, d'une carte d'identification à circuit intégré (dite "carte à puce" ou "smart card" en anglais,...). Le modèle de tâche décrit que la

10 procédure « Connect » doit être appelée lors de ce changement d'état. Cette procédure est externe au modèle de tâche, et seule une référence (ici son nom) permet de la retrouver. Le gestionnaire de tâches, suite à l'événement « connect » dans l'état « déconnecté » précise qu'il faut appeler la procédure « Connect ». Ensuite, c'est au module qui interroge le gestionnaire de tâches d'appeler cette

15 procédure. Ainsi, le modèle de tâche n'a aucune idée du contenu de cette procédure. Bien entendu, le bon sens voudrait, dans le cas présent qu'il y ait vérification du mot d'identification et du mot de passe de l'utilisateur. Si les paramètres ainsi vérifiés sont incorrects, donc refusés ("connection refused"), le gestionnaire ramène l'utilisateur dans l'état initial ("déconnecté", par le trajet en traits interrompus). Le

20 gestionnaire ne fait que répondre à des requêtes et mémoriser l'état de l'utilisateur. Il mémorise donc que le nouvel état est « déconnecté » et rend cette réponse à son appelant. Dans l'affirmative, le gestionnaire renvoie le profil fourni par la procédure d'interaction (profil qui peut par exemple figurer sur ladite carte d'identification) pour déclencher l'action de connexion et il mémorise que l'utilisateur est désormais

25 dans l'état connecté et répond cela à son appelant. Dans le présent exemple, ces paramètres sont ceux d'une liste de vol d'un utilisateur d'une compagnie aérienne ("Flight list"), et la connexion est établie avec ce profil d'utilisateur, qui passe donc dans le deuxième état.

La partie de fichier suivante correspond à la traduction de ce schéma sous

30 forme XML.

```

<state id="disconnected">
  <events>
    <event id="connect">
      <in_param id="userid" type="java.lang.String" />
      <in_param id="password" type="java.lang.String" />
      <interaction_call id="connect" >
        <method id="processing.business.handler.Connect" />
        <next_states>
          <positive>
            <out_param id="flight_list"
              type="business.FlightList" />
            <next_state id="connected" />
          </positive>
          <negative>
            <out_param id="connection_refused"
              type="java.lang.String" />
            <next_state id="disconnected" />
          </negative>
        </next_states>
      </interaction_call>
    </event>
  </events>
</state>

```

25

On a représenté en figure 2 le bloc-diagramme des principales fonctions mises en œuvre par le procédé de l'invention. Comme précisé ci-dessus, l'invention part d'un modèle de tâche augmenté (1) qui communique bi-directionnellement avec le gestionnaire de tâches (2). Ce dernier coopère avec des services (3) de l'application. Le gestionnaire de tâches fournit des services aux modules qui en ont besoin. Le diagramme de la figure 3 résume le processus de fonctionnement du

30

gestionnaire de tâches. Le processus débute par l'initialisation et le positionnement de l'état initial de l'utilisateur (4). Cette étape est suivie de la demande de la prochaine procédure d'interaction en liaison avec un événement dû à l'utilisateur (5). Cette demande dialogue avec la demande du prochain état de l'utilisateur en fonction du

5 résultat de la procédure d'interaction (6), qui dialogue de son côté avec la demande de la prochaine procédure d'interaction en fonction d'un événement externe (7).

REVENDICATIONS

1. Procédé d'augmentation d'un modèle de tâche pour permettre la gestion de l'interaction homme-machine, caractérisé en ce qu'à partir d'un modèle de tâche existant, on l'augmente avec l'état courant de l'utilisateur dans sa tâche, on décrit les événements permettant un changement d'état de l'utilisateur, et on décrit pour un événement survenu lors d'un état de l'utilisateur l'interaction à effectuer avec l'utilisateur pour gérer cet événement.
2. Procédé selon la revendication 1, caractérisé en ce que l'on ajoute avant chaque procédure d'interaction la liste des contraintes nécessaires pour déclencher l'interaction.
3. Procédé selon la revendication 1 ou 2, caractérisé en ce que l'on ajoute après chaque procédure d'interaction les valeurs que doit fournir cette interaction selon le résultat de l'interaction et qui doivent être présentées à l'utilisateur en retour.
4. Procédé selon l'une des revendications précédentes, caractérisé en ce que l'on utilise un module externe qui fournit une abstraction des actions de l'utilisateur sous forme d'événements de haut niveau.
5. Procédé selon la revendication 1 ou 2, caractérisé en ce que l'on modifie en temps réel le modèle de tâche.
6. Procédé selon l'une des revendications 1 à 3, caractérisé en ce que l'on modifie l'interaction avec l'utilisateur en temps réel.
7. Procédé selon l'une des revendications précédentes, caractérisé en ce qu'un module d'apprentissage réalise l'apprentissage à partir de l'activité d'un utilisateur, selon le modèle de tâche augmenté.
8. Procédé selon l'une des revendications précédentes, caractérisé en ce que les spécifications des services d'interface homme-machine sont dérivés du modèle de tâche augmenté.
9. Procédé selon l'une des revendications précédentes, caractérisé en ce que le modèle de tâche est fourni par un expert.

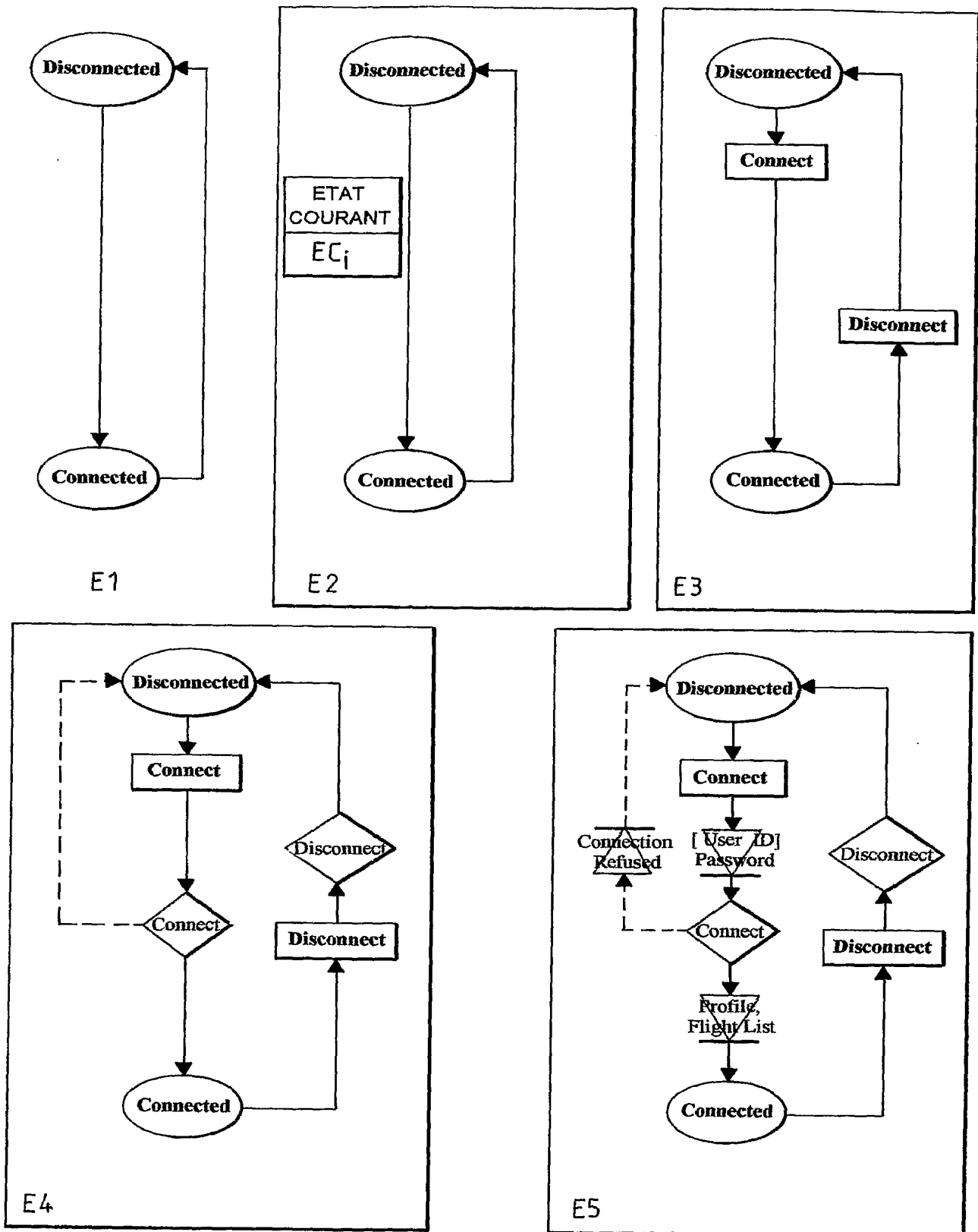


FIG.1

